

The image shows a large, symmetrical pattern composed of black text symbols on a white background. The pattern is roughly triangular in shape, oriented vertically. It features three distinct horizontal bands of symbols. The top band consists of 'SSSSSSSSSSSS' symbols. The middle band consists of 'YYY' symbols. The bottom band consists of 'SSSS' symbols. The arrangement is such that the symbols in each row of one band align with the symbols in the center of the rows of the other bands, creating a repeating, wave-like effect across the entire pattern.

\*\*\*FILE\*\*\*ID\*\*\*SYSLOAVEC

E 9

SSSSSSSS YY YY SSSSSSSS LL 000000 AAAA VV VV EEEEEEEE CCCCCCCC  
SSSSSSSS YY YY SS SS LL 000000 AA AA VV VV EE EE CC CC  
SS SS YY YY SS SS LL 000000 AA AA VV VV EE EE CC CC  
SS SS YY YY SS SS LL 000000 AA AA VV VV EE EE CC CC  
SSSSSS SSSSSS LL 000000 AA AA VV VV EEEEEEEE CC CC  
SSSSSS SS SS LL 000000 AAAA VV VV EE EE CC CC  
SS SS YY YY SS LL 000000 AAAA VV VV EE EE CC CC  
SS SS YY YY SS LL 000000 AA AA VV VV EE EE CC CC  
SSSSSSSS YY YY SSSSSSSS LLLLLLLL 000000 AA AA VV VV EEEEEEEE CCCCCCCC  
SSSSSSSS YY YY SSSSSSSS LLLLLLLL 000000 AA AA VV VV EEEEEEEE CCCCCCCC

The diagram illustrates a sequence of binary strings arranged in a grid. The strings are as follows:

- Row 1: L
- Row 2: LL
- Row 3: LLL
- Row 4: LLLL
- Row 5: LLLLL
- Row 6: LLLLLL
- Row 7: LLLLLLL
- Row 8: LLLLLLLL
- Row 9: LLLLLLLLL
- Row 10: L
- Row 11: I
- Row 12: S
- Row 13: SSS
- Row 14: SSSS
- Row 15: SSSSS
- Row 16: SSSSSS
- Row 17: SSSSSSS
- Row 18: SSSSSSSS
- Row 19: SSSSSSSSS

```
0000 1 .IF NDF_PRMSW ;FOR LINKAGE WITH SYS.EXE, ...
0000 2 :TITLE SYSLOAVEC - SYSTEM VECTORS TO LOADABLE SUBROUTINES
0000 3
0000 4 .IFF ;FOR LINKAGE WITH SYSLOAxxx.EXE, ...
0000 5 :TITLE LOAVEC - OFFSETS TO LOADABLE SUBROUTINES
0000 6 :ENDC
0000 7
0000 8 .IDENT 'V04-000'
0000 9
0000 10 ****
0000 11 *
0000 12 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 13 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 14 * ALL RIGHTS RESERVED.
0000 15 *
0000 16 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 17 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 18 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 19 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 20 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 21 * TRANSFERRED.
0000 22 *
0000 23 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 24 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 25 * CORPORATION.
0000 26 *
0000 27 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 28 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 29 *
0000 30 *
0000 31 ****
0000 32 *
0000 33 ++
0000 34 *
0000 35 :FACILITY:
0000 36 :
0000 37 : EXECUTIVE, LOADABLE SUBROUTINES
0000 38 :
0000 39 :ABSTRACT:
0000 40 :
0000 41 : VECTORS, LOAD IMAGE SIZE, AND UNDEFINED VECTOR HANDLER FOR
0000 42 : LOADABLE SUBROUTINES.
0000 43 :
0000 44 :AUTHOR:
0000 45 :
0000 46 : N. KRONENBERG, MARCH 13, 1979.
0000 47 :
0000 48 :MODIFIED BY:
0000 49 :
0000 50 : V03-021 ROW0407 Ralph O. Weber 25-JUL-1984
0000 51 : Add EXESMNTVER_DVI_ASSIST, an escape transfer vector which
0000 52 : allows $GETDVI support for shadow sets to be shipped with
0000 53 : mount verification in SYSLOAxxx.
0000 54 :
0000 55 : V03-020 WHM0001 Bill Matthews 09-Jul-1984
0000 56 : Added vectors for routines CONSINIT_CTY, CON$GETCHAR and CON$PUTCHAR
0000 57 :
```

6 9

0000	58 :	V03-019 ROW0384	Ralph O. Weber	7-JUL-1984
0000	59 :	Add the following transfer or pointer vectors for Mount Verification:		
0000	60 :	o EXESCLUTRANIO, VAXcluster state transition I/O block.		
0000	61 :	o EXESUPDGNUM, update shadow set generation number.		
0000	62 :	o EXESGL_MVMSLBA\$, MVMSL base address.		
0000	63 :			
0000	64 :			
0000	65 :	V03-018 TCM0003	Trudy C. Matthews	09-Apr-1984
0000	66 :	Add vector for routine CONSRELEASECTY.		
0000	67 :			
0000	68 :	V03-017 ROW0330	Ralph O. Weber	24-MAR-1984
0000	69 :	Add two spare mount verification transfer vectors for possible use in volume shadowing.		
0000	70 :			
0000	71 :			
0000	72 :	V03-016 KPL0001	Peter Lieberwirth	4-Mar-1984
0000	73 :	Add some extra vectors (aligned and packed) for possible use in 4.x.		
0000	74 :			
0000	75 :			
0000	76 :	V03-015 ROW0317	Ralph O. Weber	27-FEB-1984
0000	77 :	Correct patch area descriptor to have patch size in bytes not in longwords.		
0000	78 :			
0000	79 :			
0000	80 :	V03-014 ROW0292	Ralph O. Weber	4-FEB-1984
0000	81 :	Add three vectors for mount verification, which is being moved to SYSLOAxxx: EXESMOUNTVER the main entry point, EXESMNTVERSIO entry to the start I/O routine for mount verification, and EXESMNTVERSHDOL entry point to the bring shadow unit online routine (which does not exist yet). EXESMNTVERSIO allows in-driver mount verification routines to use a standard interface to starting and completing an internal I/O request. EXESMNTVERSHDOL is provided for MOUNT to use when it brings the members of a shadow set online asynchronously.		
0000	82 :			
0000	83 :			
0000	84 :			
0000	85 :			
0000	86 :			
0000	87 :			
0000	88 :			
0000	89 :			
0000	90 :			
0000	91 :	V03-013 CWH8001	CW Hobbs	4-Dec-1983
0000	92 :	Add EXESREADP_TODR and EXESWRITEP_TODR routines, used to force access of physical TODR on Nautilus CPU. On other CPUs, these routines are second labels on EXESREAD_TODR and EXESWRITE_TODR.		
0000	93 :			
0000	94 :			
0000	95 :			
0000	96 :			
0000	97 :	V03-012 KDM0062	Kathleen D. Morse	19-Jul-1983
0000	98 :	Add routine to initialize the time-wait macro data cells, EXESINI_TIMWAIT.		
0000	99 :			
0000	100 :			
0000	101 :	V03-011 KTA0001	Kerbey T. Altmann	12-Jul-1983
0000	102 :	Add routine for console device data structure initialization, INISCONSOLE.		
0000	103 :			
0000	104 :			
0000	105 :	V03-010 KDM0057	Kathleen D. Morse	12-Jul-1983
0000	106 :	Change the SYSINIT routine, SIP_SETTIME, to a loadable, CPU-dependent routine, EXESINIT_TODR.		
0000	107 :			
0000	108 :			
0000	109 :	V03-009 KDM0048	Kathleen D. Morse	07-Jun-1983
0000	110 :	Add loadable routines for referencing TODR: EXESREAD_TODR and EXESWRITE_TODR.		
0000	111 :			
0000	112 :			
0000	113 :	V03-008 WMC0001	Wayne Cardoza	09-Jun-1983
0000	114 :	Add loadable code system service dispatchers.		

0000 115 :			
0000 116 :	V03-007 JWH0203	Jeffrey W. Horn	22-Mar-1983
0000 117 :	Use SLVTAB macro to set up loadable-image header.		
0000 118 :			
0000 119 :	V03-006 TCM0002	Trudy C. Matthews	16-Feb-1983
0000 120 :	Add CONSOWNCTY vector.		
0000 121 :			
0000 122 :	V03-005 TCM0001	Trudy C. Matthews	13-Jan-1983
0000 123 :	Add SYSL\$CLRSBIA entry point. Add an alternative default		
0000 124 :	routine that RSBs harmlessly if called before code is		
0000 125 :	loaded, instead of halting.		
0000 126 :			
0000 127 :	V03-004 KTA3018	Kerbey T. Altmann	30-Oct-1982
0000 128 :	Redo to reflect changes in modules.		
0000 129 :			
0000 130 :	V03-003 MSH0001	Maryann Hinden	24-Sep-1982
0000 131 :	Change entry EXESDW780_INT to EXESUBAERR_INT.		
0000 132 :			
0000 133 :	V03-002 STJ3026	Steven T. Jeffreys	24-Sep-1982
0000 134 :	Moved LOADVEC macro to SYSMAR.		
0000 135 :			
0000 136 :	V03-001 BLS0183	Benn Schreiber	16-Aug-1982
0000 137 :	Add new entry points for loadable console support		
0000 138 :			
0000 139 :--			

0000 141 :♦  
0000 142 : THIS MODULE MAY BE ASSEMBLED WITH EXECUTIVE MODULE PRMSW.MAR  
0000 143 : WHICH DEFINES THE CONDITIONAL CONTROL VARIABLE PRMSW. IF PRMSW  
0000 144 : IS NOT DEFINED, THEN SYSLOAVEC.MAR IS INTENDED TO LINK WITH SYS.EXE.  
0000 145 : IF PRMSW IS DEFINED, SYSLOAVEC.MAR IS INTENDED FOR LINKAGE WITH  
0000 146 : SYSLOAxxx.EXE. (xxx IS THE CPU DESIGNATOR, E.G., 780.)  
0000 147 :  
0000 148 : IF PRMSW IS NOT DEFINED (LINK WITH SYS.EXE), THE SOURCE CONTAINS  
0000 149 : A LIST OF VECTORS TO BE CONNECTED TO THE SUBROUTINES AND DATA STRUCTURES  
0000 150 : IN SYSLOAxxx.EXE.  
0000 151 :  
0000 152 : DATA VECTORS ARE LONGWORD POINTERS TO THE LOADED DATA STRUCTURES. BEFORE  
0000 153 : LOADING, THE POINTERS CONTAIN A 0.  
0000 154 :  
0000 155 : SUBROUTINE VECTORS ARE OF THE FORM:  
0000 156 :  
0000 157 : ENTRY:: JMP 2#EXESLOAD\_ERROR  
0000 158 :  
0000 159 : THERE ARE TWO TYPES OF ROUTINE ENTRIES. THE FIRST TYPE IS LONGWORD  
0000 160 : ALIGNED ENTRIES FOR SCB VECTORS. THE SECOND TYPE IS PACKED  
0000 161 : ENTRIES FOR ORDINARY ROUTINES.  
0000 162 :  
0000 163 : SYSLOAVEC ALSO CONTAINS THE LOAD ERROR HANDLER, EXESLOAD\_ERROR,  
0000 164 : WHICH IS SIMPLY A HALT. IF ANY LOADABLE SUBROUTINE IS CALLED  
0000 165 : PRIOR TO BEING LOADED, THE HALT WILL BE EXECUTED.  
0000 166 :  
0000 167 : IF PRMSW IS DEFINED (LINK WITH SYSLOAxxx.EXE), THE SOURCE  
0000 168 : CONTAINS LONGWORD EXESLOAD\_SIZE, THE NUMBER OF BYTES IN SYSLOAxxx.EXE.  
0000 169 : THE SOURCE ALSO CONTAINS A LIST OF THE SELF-RELATIVE OFFSETS TO  
0000 170 : THE SUBROUTINES IN SYSLOAxxx.EXE. THE LIST OF OFFSETS IS USED  
0000 171 : TO CONNECT THE SYS.EXE VECTORS TO THE LOADED SUBROUTINES.  
0000 172 :-

59

52

47  
5447  
5247  
5947  
52

```

0000 174 :
0000 175 : MISC. LABELS AND DATA:
0000 176 :
0000 177 :
0000 178 .IF DF,PRMSW ;FOR LINKAGE WITH SYSLOAxx.EXE,....
0000 179 $DYNDEF
0000 180
0000 181
0000 182 .PSECT __LOAD_END,QUAD ; Force label to be at the end of
0000 183 SYSL$END:: loadable code
0000 184
0000 185 .PSECT $$S000 ; Force return to start of loadable code
0000 186 SYSL$BEGIN:::
0000 187 SLVTAB END = SYSL$END, -
0000 188 INITRTN = INI$IMAP, -
0000 189 FACILITY= <SYSLOAVEC>
0000 190 :
0000 191 : NOTE: The modules, INIADPxxx, understand how to unload the vectors
0000 192 : for routines that are only called once during initialization
0000 193 : of the system. Therefore, if the format of these vectors
0000 194 : change, then the code in INIADPxxx must also change.
0000 195 :
0000 196 :
0000 197 :
0000 198 .IFF ; For linkage with SYS.EXE,....
0000 199 .PSECT $$S500,LONG
0000 200 .ALIGN LONG ; Start vector list on longwd
0000 201 EXESAL_LOAVEC:: ; Addr of start of vectors
0000 202 .ENDC
0000 203
0000 204
0000 205 :
0000 206 : VECTOR LIST:
0000 207 : Define longword-aligned routine vectors.
0000 208 :
0000 209 :
0000 210 LOADVEC EXESMCHK,TYPE=2 ;MACHINE CHECK HANDLER
0006 211 LOADVEC EXESINT54,TYPE=2 ;INTERRUPT, SCB VECTOR=^X54
000E 212 LOADVEC EXESINT58,TYPE=2 ;INTERRUPT, SCB VECTOR=^X58
0016 213 LOADVEC EXESINT5C,TYPE=2 ;INTERRUPT, SCB VECTOR=^X5C
001E 214 LOADVEC EXESINT60,TYPE=2 ;INTERRUPT, SCB VECTOR=^X60
0026 215 LOADVEC UBA$UNEXINT,TYPE=2, - ;INTERRUPT, UNIBUS ITSELF
0026 216 SEC_LABEL=UBA_UNEXINT
002E 217 :
002E 218 : Extra aligned vectors. Current target of these vectors in SYSLOA is a
002E 219 : halt instruction in ERRSUB*.
002E 220 :
002E 221 LOADVEC EXESEXTRA1,TYPE=2 ;EXTRA LABEL
0036 222 LOADVEC EXESEXTRA2,TYPE=2 ;EXTRA LABEL
003E 223 LOADVEC EXESEXTRA3,TYPE=2 ;EXTRA LABEL
0046 224 LOADVEC EXESEXTRA4,TYPE=2 ;EXTRA LABEL
004E 225 LOADVEC EXESEXTRA5,TYPE=2 ;EXTRA LABEL
0056 226 :
0056 227 : Define packed routine vectors.
0056 228 :
0056 229 :
0056 230 LOADVEC ECC$REENABLE ;MEMORY ERROR TIMERS

```

005C	231	LOADVEC EXESINIBOOTADP	:INIT BOOT DEVICE ADAPTER
0062	232	LOADVEC EXESDUMPCCPUREG	:DUMP CPU-SPECIFIC IPR'S INTO EMB
0068	233	LOADVEC EXESREGRESTOR	:RESTORE CPU-SPECIFIC IPR'S
006E	234	LOADVEC EXESREGSAVE	:SAVE CPU-SPECIFIC IPR'S
0074	235	LOADVEC EXESINIPROCREG	:INIT PROCESSOR REGISTERS
007A	236	LOADVEC EXESTEST_CSR	:TEST UB CSR FOR EXISTENCE
0080	237	LOADVEC IOCSPURGdatap	:PURGE DATAPATH
0086	238	LOADVEC INISMPMADP	:INITIALIZE MULTIPORT MEMORY
008C	239	LOADVEC EXESSSTARTUPADP	:Startup up any adapters
0092	240	LOADVEC EXESSHUTDWNADP	:Shutdown any (all) adapters
0098	241	LOADVEC MASRAVAIL	:Multiport memory
009E	242	LOADVEC MASREQUEST	:Multiport memory request
00A4	243	LOADVEC MASINITIAL	:Multiport initialization
00AA	244	LOADVEC CONSSTARTIO	:Console start I/O
00B0	245	LOADVEC CONSSET LINE	:Set line
00B6	246	LOADVEC CONSDS SET	:Data set
00BC	247	LOADVEC CONSXON	:XON to console
00C2	248	LOADVEC CONSXOFF	:XOFF to console
00C8	249	LOADVEC CONSSTOP	:STOP output
00CE	250	LOADVEC CONSSTOP2	:stop for 2 seconds
00D4	251	LOADVEC CONSABORT	:Abort console I/O
00DA	252	LOADVEC CONSRESUME	:Resume output
00E0	253	LOADVEC CONSSET MODEM	:Set modem
00E6	254	LOADVEC CONSNULL	:Null routine
00EC	255	LOADVEC CONSDISCONNECT	:Console disconnect routine
00F2	256	LOADVEC CONSINITIAL	:Initialize console controller
00F8	257	LOADVEC CONSINITLINE	:Initialize console line
00FE	258	LOADVEC CONSINTINTP	:Input interrupt
0104	259	LOADVEC CONSINTOUT	:Output interrupt
010A	260	LOADVEC CONSSENDCONSCMD	:Send cpu-dependent command to console
0110	261	LOADVEC SYSLSCLRSBIA, -	:Clear SBIA error bits
0110	262	DEF RTN=EXESLOAD_NOP	
0116	263	LOADVEC CONSOWNCTY	:Set up to talk directly to console tty
011C	264	LOADVEC CONSRELEASECTY	:Restore normal console cty interface
0122	265	LOADVEC CONSGETCHAR	:Get a character from the console cty
0128	266	LOADVEC CONSPUTCHAR	:Put a character out to the console cty
012E	267	LOADVEC CONSINIT_CTY	:Initialization routine for the console cty
0134	268	LOADVEC EXESREAD_TODR	:Read Time-Of-Day Register
013A	269	LOADVEC EXESWRITE_TODR	:Write Time-Of-Day Register
0140	270	LOADVEC EXESINIT_TODR	:Initialize system time-of-day
0146	271	LOADVEC INISCONSOLE	:Initialize console device data struc
014C	272	LOADVEC EXESINI_TIMWAIT	:Initialize time-wait loop data cells
0152	273	LOADVEC EXESREADP_TODR	:Read physical TODR register
0158	274	LOADVEC EXESWRITEP_TODR	:Write physical TODR register
015E	275	LOADVEC EXESMNTVER	:Mount verification main entry point
0164	276	LOADVEC EXESMNTVERSIO	:Mount verification start I/O request
016A	277	LOADVEC EXESMNTVERSHDOL	:Mount verification online shadow unit
0170	278	LOADVEC EXESCLUTRANIO	:Mount verification VAXcluster state transition block I/O
0176	279	LOADVEC EXESUPDGNERNUM	:Mount verification update shadow set generation number
017C	280		
017C	281		
017C	282	LOADVEC EXESMNTVER_DVI_ASSIST	:Mount verification \$GETDVI escape
0182	283	LOADVEC EXESMNTVERSP1	:Mount verification spare xfer vector
0188	284	LOADVEC EXESMNTVERSP2	:Mount verification spare xfer vector
018E	285	LOADVEC EXESGL_MVMSLBAS, 1.. -	:Mount verification MVMSL base address
018E	286	EXESAB_MVMSLBAS	
0194	287	:	

```

0194 288 : Extra packed vectors. Current target of these vectors in SYSLOA is a
0194 289 : halt instruction in ERRSUB*.
0194 290 :
0194 291 LOADVEC EXESEXTRA6 :EXTRA LABEL
019A 292 LOADVEC EXESEXTRA7 :EXTRA LABEL
01A0 293 LOADVEC EXESEXTRA8 :EXTRA LABEL
01A6 294 LOADVEC EXESEXTRA9 :EXTRA LABEL
01AC 295 LOADVEC EXESEXTRA10 :EXTRA LABEL
01B2 296
01B2 297 :
01B2 298 : Define pointers to data structures.
01B2 299 :
01B2 300
01B2 301 LOADVEC EXESMCHK_ERRCNT,TYPE=1 ;Point to array of mchk error counters.
01B8 302
01B8 303 .IF DF,PRMSW
01B8 304 .LONG -1 ; For the loadable image
01B8 305 .ENDC ; put in a stopper signal
01B8 306
01B8 307 :
01B8 308 : IF LINKAGE WITH SYS.EXE, DEFINE A LOAD ERROR HANDLER AND A HANDLER THAT
01B8 309 : RSB'S HARMLESSLY (FOR A ROUTINE USED BY XDELTA THAT MAY BE CALLED BEFORE
01B8 310 : BEING LOADED).
01B8 311 :
01B8 312 :
01B8 313
01B8 314 .IF NDF,PRMSW
00 01B8 315 EXESLOAD_ERROR:: ;COME HERE IF CALL TO UNLOADED
01B8 316 HALT ;SUBROUTINE
05 01B9 317 EXESLOAD_NOP:: ;COME HERE IF ROUTINE NOT LOADED
01BA 318 RSB ;BUT DON'T WANT TO ERROR HALT
01BA 319 .ENDC
01BA 320
01BA 321 :
01BA 322 : IF LINKAGE WITH SYS.EXE, DEFINE THE DISPATCH VECTOR FOR LOADABLE CODE
01BA 323 : SYSTEM SERVICE DISPATCHERS. THERE ARE SEPARATE VECTORS FOR EXEC AND
01BA 324 : KERNEL MODE WITH TWO SPARE ENTRIES IN EACH.
01BA 325 :
01BA 326
01BA 327 .IF NDF,PRMSW
01BA 328
01BA 329 EXESLOAD_KDISP:: ;Kernel mode dispatchers
01BA 330 EXESLOAD_KCJF:: ;CJF
000001B9'9F 16 01BA 331 JSB @#EXESLOAD_NOP ;CJF
000001B9'9F 16 01C0 332 EXESLOAD_KRUF:: ;RUF
000001B9'9F 16 01C6 333 JSB @#EXESLOAD_NOP ;RUF
000001B9'9F 16 01CC 334 EXESLOAD_KSPR1:: ;First spare
000001B9'9F 16 01CC 335 JSB @#EXESLOAD_NOP ;First spare
000001B9'9F 16 01D2 336 EXESLOAD_KSPR2:: ;Second spare
000001B9'9F 16 01D2 337 JSB @#EXESLOAD_NOP ;Second spare
000001B9'9F 16 01D3 338 RSB
000001B9'9F 16 01D3 339
000001B9'9F 16 01D3 340 EXESLOAD_EDISP:: ;Exec mode dispatchers
000001B9'9F 16 01D3 341 EXESLOAD_ESPR1:: ;First spare
000001B9'9F 16 01D9 342 JSB @#EXESLOAD_NOP ;First spare
000001B9'9F 16 01D9 343 EXESLOAD_ESPR2:: ;Second spare
000001B9'9F 16 01D9 344 JSB @#EXESLOAD_NOP ;Second spare

```

05 01DF 345 RSB  
01E0 346  
01E0 347 .ENDC  
01E0 348  
01E0 349 :  
01E0 350 : IF LINKAGE WITH SYSLOAXXX.EXE, DEFINE 15 LONGWORDS OF PATCH AREA:  
01E0 351 :  
01E0 352 :  
01E0 353 .IF DF\_PRMSW  
01E0 354 .PSECT \_PATCH  
01E0 355 PATCH\_DESC::  
01E0 356 .LONG 15\*4  
01E0 357 .LONG PATCH\_AREA  
01E0 358 PATCH\_AREA::  
01E0 359 .BLKL 15  
01E0 360 .ENDC  
01E0 361  
01E0 362 .END

CONSABORT	000000D4	RG	01	EXESMNTVERSHDOL	0000016A	RG	01
CONSDISCONNECT	000000EC	RG	01	EXESMNTVERSIO	00000164	RG	01
CONSDS_SET	000000B6	RG	01	EXESMNTVERSP1	00000182	RG	01
CONSGETCHAR	00000122	RG	01	EXESMNTVERSP2	00000188	RG	01
CONSINITIAL	000000F2	RG	01	EXESMNTVER_DVI_ASSIST	0000017C	RG	01
CONSINITLINE	000000F8	RG	01	EXESMOUNTVER	0000015E	RG	01
CONSINIT_CTY	0000012E	RG	01	EXESREADP TODR	00000152	RG	01
CONSINITNP	000000FE	RG	01	EXESREAD TODR	00000134	RG	01
CONSINTOUT	00000104	RG	01	EXESREGRESTOR	00000068	RG	01
CONSNULL	000000E6	RG	01	EXESREGSAVE	0000006E	RG	01
CONSOWNCTY	00000116	RG	01	EXESSHUTDWNADP	00000092	RG	01
CONSPUTCHAR	00000128	RG	01	EXESSSTARTUPADP	0000008C	RG	01
CONSRELEASECTY	0000011C	RG	01	EXESTEST_CSR	0000007A	RG	01
CONSRESUME	000000DA	RG	01	EXESUPDGNUM	00000176	RG	01
CONSSENDCONSCMD	0000010A	RG	01	EXESWRITEP TODR	00000158	RG	01
CONSSET_LINE	000000B0	RG	01	EXESWRITE TODR	0000013A	RG	01
CONSSET_MODEM	000000E0	RG	01	INISCONSOLE	00000146	RG	01
CONSSTARTIO	000000AA	RG	01	INISMPMAFP	00000086	RG	01
CONSSTOP	000000C8	RG	01	IOCSPURGDATAP	00000080	RG	01
CONSSTOP2	000000CE	RG	01	MASINITIAL	000000A4	RG	01
CONSXOFF	000000C2	RG	01	MASRAVAIL	00000098	RG	01
CONSXON	000000BC	RG	01	MASREQUEST	0000009E	RG	01
ECC\$REENABLE	00000056	RG	01	SYSLSCLRSBIA	00000110	RG	01
EXESAL_LOAVEC	00000000	RG	01	UBASUNEXT	00000028	RG	01
EXESCLUTRANIO	00000170	RG	01				
EXESDUMPPCPUREG	00000062	RG	01				
EXESEXTRA1	00000030	RG	01				
EXESEXTRA10	000001AC	RG	01				
EXESEXTRA2	00000038	RG	01				
EXESEXTRA3	00000040	RG	01				
EXESEXTRA4	00000048	RG	01				
EXESEXTRA5	00000050	RG	01				
EXESEXTRA6	00000194	RG	01				
EXESEXTRA7	0000019A	RG	01				
EXESEXTRA8	000001A0	RG	01				
EXESEXTRA9	000001A6	RG	01				
EXESGL_MVMSLBAS	00000190	RG	01				
EXESINITBOOTADP	0000005C	RG	01				
EXESINIPROCREG	00000074	RG	01				
EXESINIT_TODR	00000140	RG	01				
EXESINI_TIMWAIT	0000014C	RG	01				
EXESINT54	00000008	RG	01				
EXESINT58	00000010	RG	01				
EXESINT5C	00000018	RG	01				
EXESINT60	00000020	RG	01				
EXESLOAD_EDISP	000001D3	RG	01				
EXESLOAD_ERROR	000001B8	RG	01				
EXESLOAD_ESPR1	000001D3	RG	01				
EXESLOAD_ESPR2	000001D9	RG	01				
EXESLOAD_KCJF	000001BA	RG	01				
EXESLOAD_KDISP	000001BA	RG	01				
EXESLOAD_KRUF	000001C0	RG	01				
EXESLOAD_KSPR1	000001C6	RG	01				
EXESLOAD_KSPR2	000001CC	RG	01				
EXESLOAD_NOP	000001B9	RG	01				
EXESMCHK	00000000	RG	01				
EXESMCHK_ERRCNT	000001B4	RG	01				

+-----+  
 ! Psect synopsis !  
 +-----+

## PSECT name

	Allocation	PSECT No.	Attributes
ABS .	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$SS500	000001E0 ( 480.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+  
 ! Performance indicators !  
 +-----+

## Phase

	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.10	00:00:01.49
Command processing	125	00:00:00.57	00:00:05.31
Pass 1	153	00:00:05.26	00:00:16.05
Symbol table sort	0	00:00:00.06	00:00:00.06
Pass 2	83	00:00:01.57	00:00:05.18
Symbol table output	10	00:00:00.08	00:00:00.22
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	411	00:00:07.67	00:00:28.34

The working set limit was 1200 pages.

25770 bytes (51 pages) of virtual memory were used to buffer the intermediate code.

There were 10 pages of symbol table space allocated to hold 81 non-local and 0 local symbols.

362 source lines were read in Pass 1, producing 15 object records in Pass 2.

2 pages of virtual memory were used to define 1 macro.

+-----+  
 ! Macro library statistics !  
 +-----+

## Macro library name

	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB:1	1
\$255\$DUA28:[SYSLIB]STARLET.MLB:2	0
TOTALS (all libraries)	1

58 GETS were required to define 1 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:SYSLOAVEC/OBJ=OBJ\$:SYSLOAVEC MSRC\$:SYSLOAVEC/UPDATE=(ENHS:SYSLOAVEC)+EXECMLS/LIB

0386 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

SYSARAM  
LIS

SYSMTRACE  
LIS

SYSIMGSTA  
LIS

SYSLHM  
LIS

SYSLOGNAME  
LIS

SYSLKSET  
LIS

SYSMAILBX  
LIS